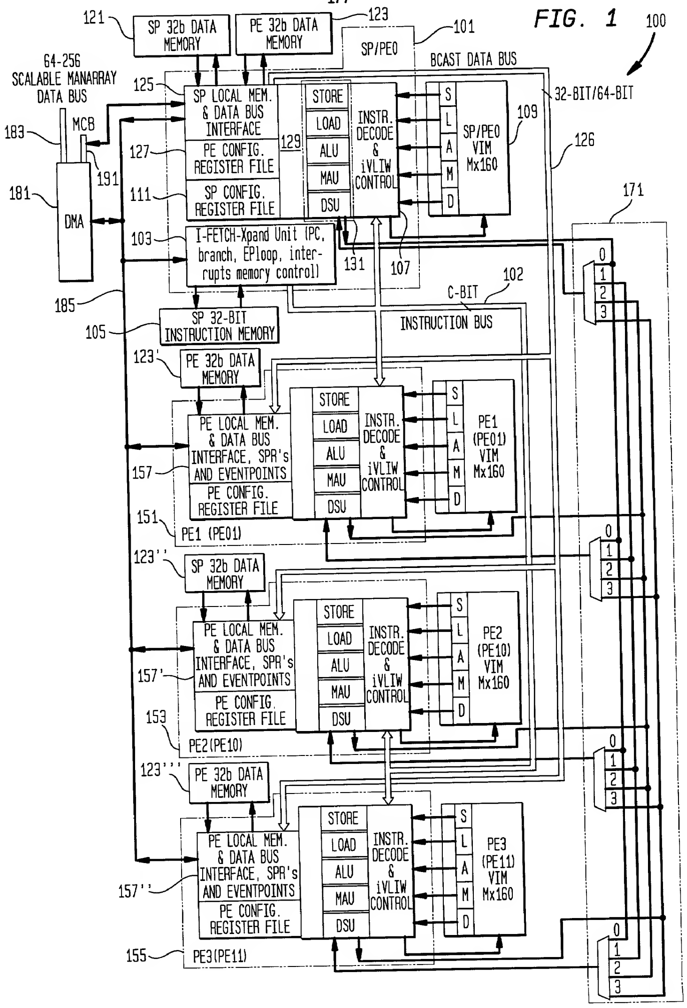
FIG. 1
100
121
SP 32b DATA MEMORY
PE 32b DATA MEMORY
123
SP/PE0
101
BCAST DATA BUS
32-BIT/64-BIT
64-256 SCALABLE MANARRAY DATA BUS
125
SP LOCAL MEM. & DATA BUS INTERFACE
STORE
LOAD
ALU
MAU
DSU
INSTR. DECODE & iVLIW CONTROL
S
L
A
M
D
SP/PE0 VIM Mx160
109
126
183
MCB
127
PE CONFIG. REGISTER FILE
129
191
181
111
SP CONFIG. REGISTER FILE
DMA
171
103
I-FETCH-Xpand Unit (PC, branch, EPloop, interrupts memory control)
131
107
0
1
2
3
102
C-BIT
185
105
SP 32-BIT INSTRUCTION MEMORY
INSTRUCTION BUS
123'
PE 32b DATA MEMORY
PE LOCAL MEM. & DATA BUS INTERFACE, SPR's AND EVENTPOINTS
PE1 (PE01) VIM Mx160
157
151
PE1 (PE01)
123''
SP 32b DATA MEMORY
157'
PE2 (PE10) VIM Mx160
153
PE2(PE10)
123'''
157''
PE3 (PE11) VIM Mx160
155
PE3(PE11)

FIG. 2A

200

LV Encoding

31 30	29	28 27 26 25	24	23 22	21 20 19 18	17	16	15	14	13	12	11	10	9	8	7 6 5 4 3 2 1 0
Group	S/P	CtrlOp	E/D	UAF	InstrCnt	0	0	0	SU	LU	ALU	MAU	DSU	Vb	0	VIMOFFS

FIG. 2B

210

LV Syntax/Operation

```
Instruction Operands          Operation
LV.[SP]     V[01], VIMOFFS,   (V[01]+VIMOFFS)[SU].enable ← 0 if (D = S)
            InstrCnt,         (V[01]+VIMOFFS)[LU].enable ← 0 if (D = L)
            D = {SLAMD},      (V[01]+VIMOFFS)[ALU].enable ← 0 if (D = A)
            F = [AMDN]        (V[01]+VIMOFFS)[MAU].enable ← 0 if (D = M)
                              (V[01]+VIMOFFS)[DSU].enable ← 0 if (D = D)

                              (V[01]+VIMOFFS)[UAF] ← ALU if (F = A or F =)
                              (V[01]+VIMOFFS)[UAF] ← MAU if (F = M)
                              (V[01]+VIMOFFS)[UAF] ← DSU if (F = D)
                              (V[01]+VIMOFFS)[UAF] ← None if (F = N)

                              for (i = 0;i < InstrCnt;i++){
                                 Load instruction into (V[01]+VIMOFFS)
                                 if (SU Instr AND D! = S){(V[01]+VIMOFFS)[SU].enable ← 1}
                                 if (LU Instr AND D! = L){(V[01]+VIMOFFS)[LU].enable ← 1}
                                 if (ALU Instr AND D! = A){(V[01]+VIMOFFS)[ALU].enable ← 1}
                                 if (MAU Instr AND D! = M){(V[01]+VIMOFFS)[MAU].enable ← 1}
                                 if (DSU Instr AND D! = D){(V[01]+VIMOFFS)[DSU].enable ← 1}
                              }
```

FIG. 3A

300

XV Encoding

31 30	29	28 27 26 25	24	23 22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7 6 5 4 3 2 1 0
Group	S/P	CtrlOp	VX	UAF	0	0	0	0	0	0	0	SU	LU	ALU	MAU	DSU	Vb	0	VimOffs

FIG. 3B

310

XV Syntax/Operation

Instruction	Operands	Operation

XV.[SP]	V[01], VIMOFFS,
	E = {SLAMD}, F = [AMDN]
		Execute(V[01]+VIMOFFS)[SU] if (E = S)
		Execute(V[01]+VIMOFFS)[LU] if (E = L)
		Execute(V[01]+VIMOFFS)[ALU] if (E = A)
		Execute(V[01]+VIMOFFS)[MAU] if (E = M)
		Execute(V[01]+VIMOFFS)[DSU] if (E = D)
		
		(V[01]+VIMOFFS)[UAF] ← ALU if (F = or F = A)
		(V[01]+VIMOFFS)[UAF] ← MAU if (F = M)
		(V[01]+VIMOFFS)[UAF] ← DSU if (F = D)
		(V[01]+VIMOFFS)[UAF] ← None if (F = N)